第十一周周汇报

按照上周的计划安排，这周研读的论文是由齐国君教授写的[Loss-Sensitive GAN](https://arxiv.org/abs/1701.06264)。这篇文章算是自己开创了一个新的GAN领域，不过创作的初衷也主要是为了解决传统GAN存在的在判别器训练很好的情况下，生成器的梯度会发生消失的问题。LS-GAN在文章中分别对比了传统GAN和WGAN，并且在WGAN的基础上对一些公式做了定量分析和推证，这样让LS-GAN的思想更加的清晰，但是大量的数学公式也是让人读得吃力。

文章指出传统GAN的作者Goodfellow做了一个很大胆的假设：用来评估样本真实度的Discriminator网络具有无限的建模能力，也就是说不管真实样本和生成的样本有多复杂，判别器都能把它们区分开，这个假设也称为非参数假设。在WGAN中已经推证了在高维样本中，生成样本和真实样本之间的重叠没有即使存在也可以忽略不计由此带来的就是生成器的损失函数中JS散度为一个常数，从而导致损失函数的值为0引起梯度消失的情况。

无限建模能力是造成梯度消失得一个重要的问题，作者围绕这个思想创造了LS-GAN，直接通过限定GAN的建模能力，得到一种新的GAN模型。这篇文章的数学公式以及推导也是十分的繁琐和复杂，我就在已有数学的基础上对一些公式做出分析。公式1为LS-GAN的目标函数，这个是用来学习损失函数的目标函数，通过最小化这个目标得到一个“损失函数" ，在真实样本上越小越好，在生成的样本上越大越好。

 (1)

对应的G-网络，通过最小化公式2训练网络：

 (2)

分析上述公式，对的学习目标 S中的第二项，它是以真实样本x和生成样本的一个度量为各自的目标间隔从而把x和分开。这样的好处是如果生成的样本和真实样本已经很接近，那么就不必要求他们的非得有个固定间隔，因为这个时候生成的样本已经非常好了，已经接近或者达到了真实样本水平。此时LS-GAN就可以集中力量提高那些距离真实样本还很远，真实度不那么高的样本上了。这样就可以更合理使用LS-GAN的建模能力，一旦限定了建模能力后，也不用担心模型的生成能力有损失了，文章称之为“按需分配”。

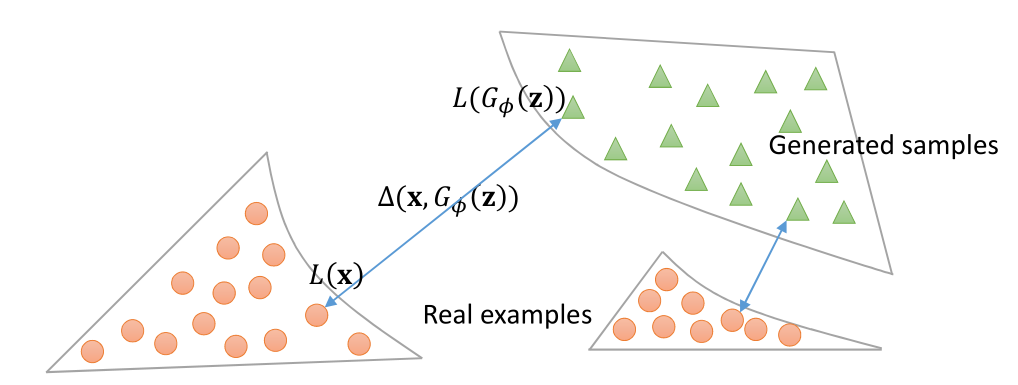
图1展示了LS-GAN的按需分配的情况，由图1我们可以看到当生成的样本绿色的点和真实的样本红色的点分布上在某一部分很相近时，通过目标函数的作用，将减小离真实样本很近的那一部分分布的训练从而将训练集中在离真实样本较远分布的生成样本的训练上，这样的“按需分配”将大大提高训练的有效性。

图1.LS-GAN“按需分配”图

文章的一个重点就是强调将LS-GAN限定在Lipschitz密度上，同时限制住的建模能力到Lipschitz连续的函数类上，从而实现了了LS-GAN得到的生成样本密度与真实密度的一致性，这一块我将在附录中简单证明一下。所谓的Lipschitz密度就是要求真实的密度分布不能变化的太快，密度的变化随着样本的变化不能无限地大，要有个度，不过这个度可以非常非常地大，只要不是无限大就好。文章用了大量的笔墨在证明LS-GAN在梯度消失上的优越性，但是推证的公式我不是很理解，但是通过作者的阐述知道：的上下界是分段线性的，这种分段线性的函数几乎处处存在非消失的梯度，这样适当地控制地学习过程，在这两个上下界之间最优是不会出现饱和现象从而达到梯度不消失的情况。

LS-GAN和GAN一样是一种无监督的学习算法，对于LS-GAN通过定义适当的损失函数，它可以非常容易的推广到有监督和半监督的学习模型。比如，我们可以定义一个有条件的损失函数，这个条件y可以是输入样本x的类别。当类别和样本一致的时候，这个损失函数会比类别不一致的时候小，文章给的损失函数为公式3，加入了标签y。

 (3)

一旦得到损失函数，在给定一个样本x后，就可以用最小化损失函数的那个类别来对样本进行分类。

LS-GAN的内容充斥着大量的数学公式，在大体上对文章有了一定的解读但是一些繁琐的数学公式还是有一些没有整理清楚，以后随着数学和编程能力的提高再去研读吧。本周把GAN的文章整理了一下，发现有很多好的创新点的GAN的文章，按照上周计划我还是把WGAN做了个后续扫尾工作把[WGAN-GP](https://arxiv.org/abs/1704.00028)文章简单的研读了一下。

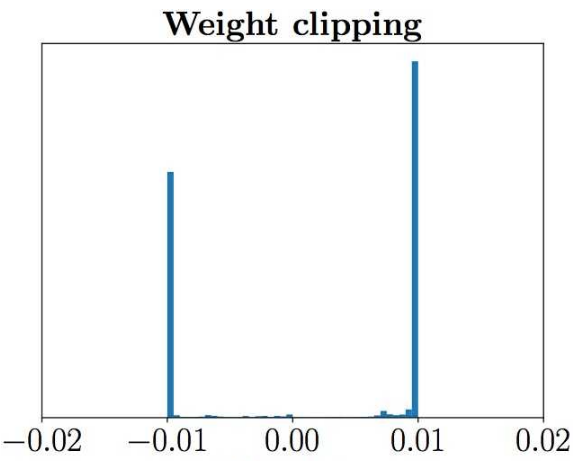
WGAN-GP是针对WGAN的存在的问题提出来的，WGAN在真实的实验过程中依旧存在着训练困难、收敛速度慢的问题，相比较传统GAN在实验上提升不是很明显。WGAN-GP在文章中指出了WGAN存在问题的原因，那就是WGAN在处理Lipschitz限制条件时直接采用了weight clipping，就是每当更新完一次判别器的参数之后，就检查判别器的所有参数的绝对值有没有超过一个阈值，比如0.01，有的话就把这些参数clip回 [-0.01, 0.01] 范围内。通过在训练过程中保证判别器的所有参数有界，就保证了判别器不能对两个略微不同的样本在判别上不会差异过大，从而间接实现了Lipschitz限制。实际训练上判别器loss希望尽可能拉大真假样本的分数差，然而weight clipping独立地限制每一个网络参数的取值范围，在这种情况下最优的策略就是尽可能让所有参数走极端，要么取最大值（如0.01）要么取最小值（如-0.01），文章通过实验验证了猜测如图2所示判别器的参数几乎都集中在最大值和最小值上。

图2.weight clipping后参数分布

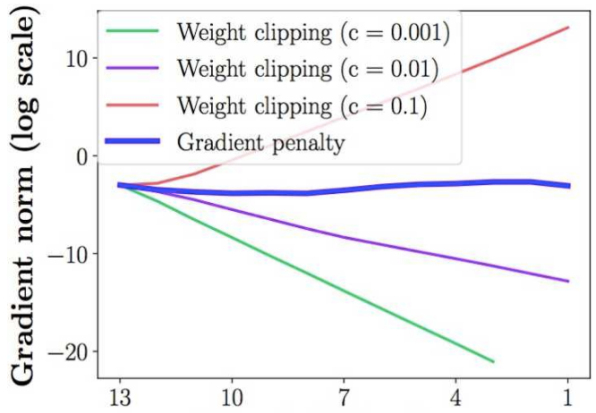
还有就是weight clipping会导致很容易一不小心就梯度消失或者梯度爆炸。原因是判别器是一个多层网络，如果把clipping threshold设得稍微小了一点，每经过一层网络，梯度就变小一点点，多层之后就会指数衰减；反之，如果设得稍微大了一点，每经过一层网络，梯度变大一点点，多层之后就会指数爆炸。只有设得不大不小，才能让生成器获得恰到好处的回传梯度，然而在实际应用中这个平衡区域可能很狭窄，就会给调参工作带来麻烦。文章也通过实验展示了这个问题，图3中横轴代表判别器从低到高第几层，纵轴代表梯度回传到这一层之后的尺度大小（注意纵轴是对数刻度）

图3.WGAN中判别器梯度变化

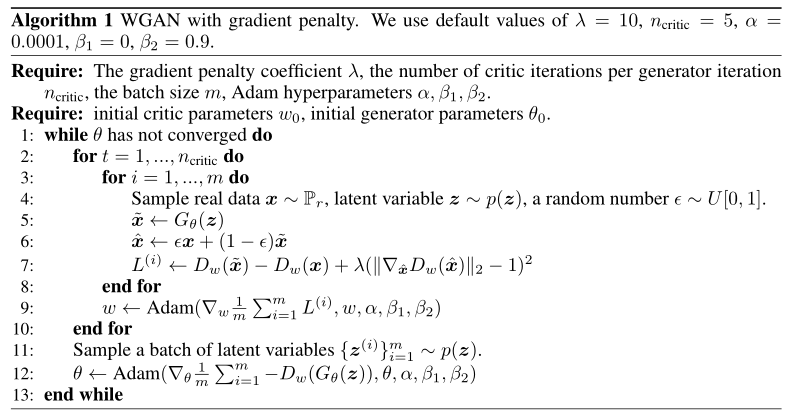
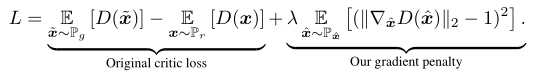
在以上问题提出后，作者提出了解决方案，那就是gradient penalty我翻译为梯度惩罚。Lipschitz限制是要求判别器的梯度不超过K，gradient penalty就是设置一个额外的loss项来实现梯度与K之间的联系，这就是gradient penalty的核心所在，图4为引入gradient penalty后WGAN-GP的算法框图，对于算法的分析我在附录中加以说明。

图4.WGAN-GP算法框图

公式4展示为WGAN-GP的最终目标函数：

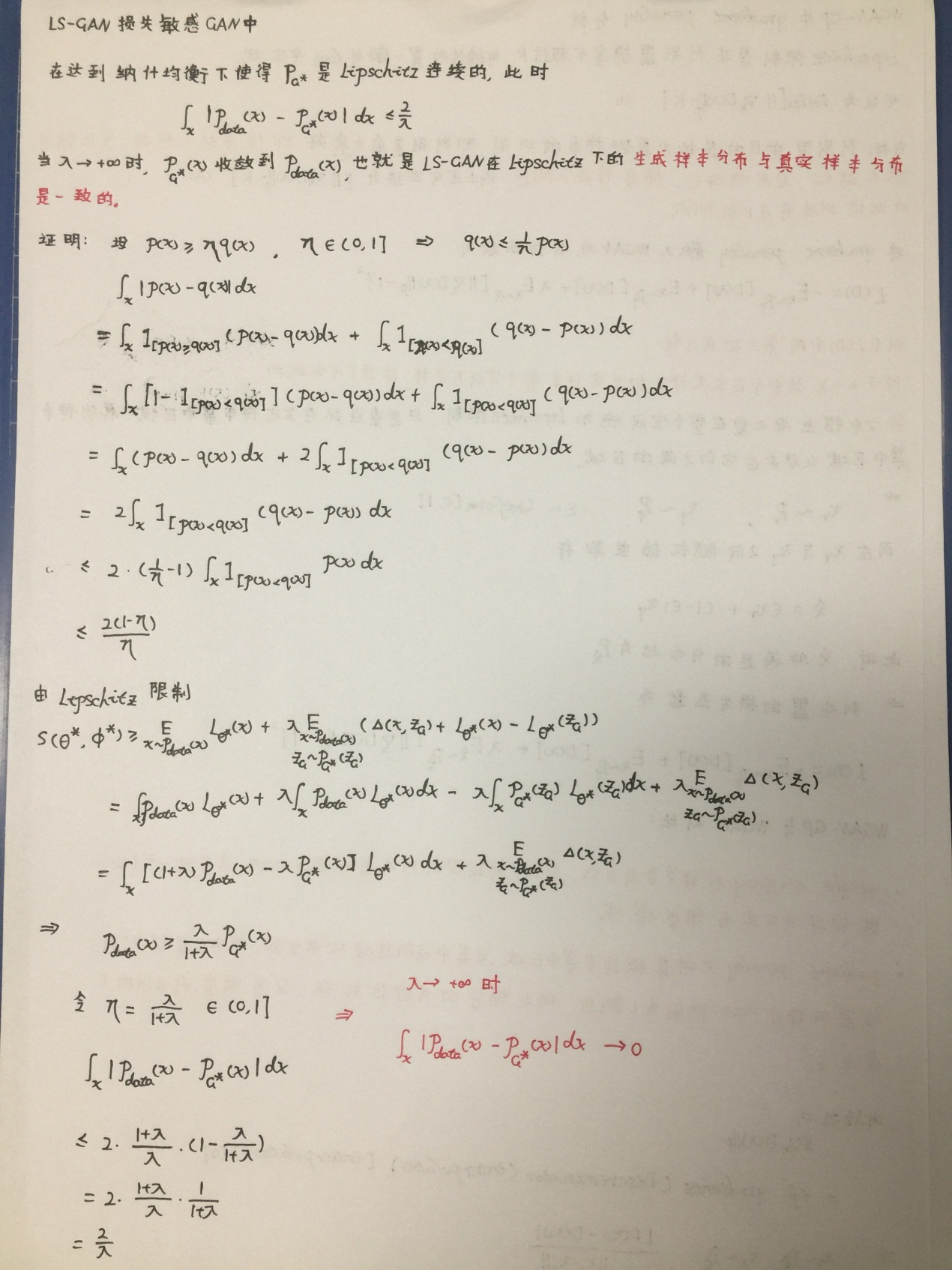
 (4)

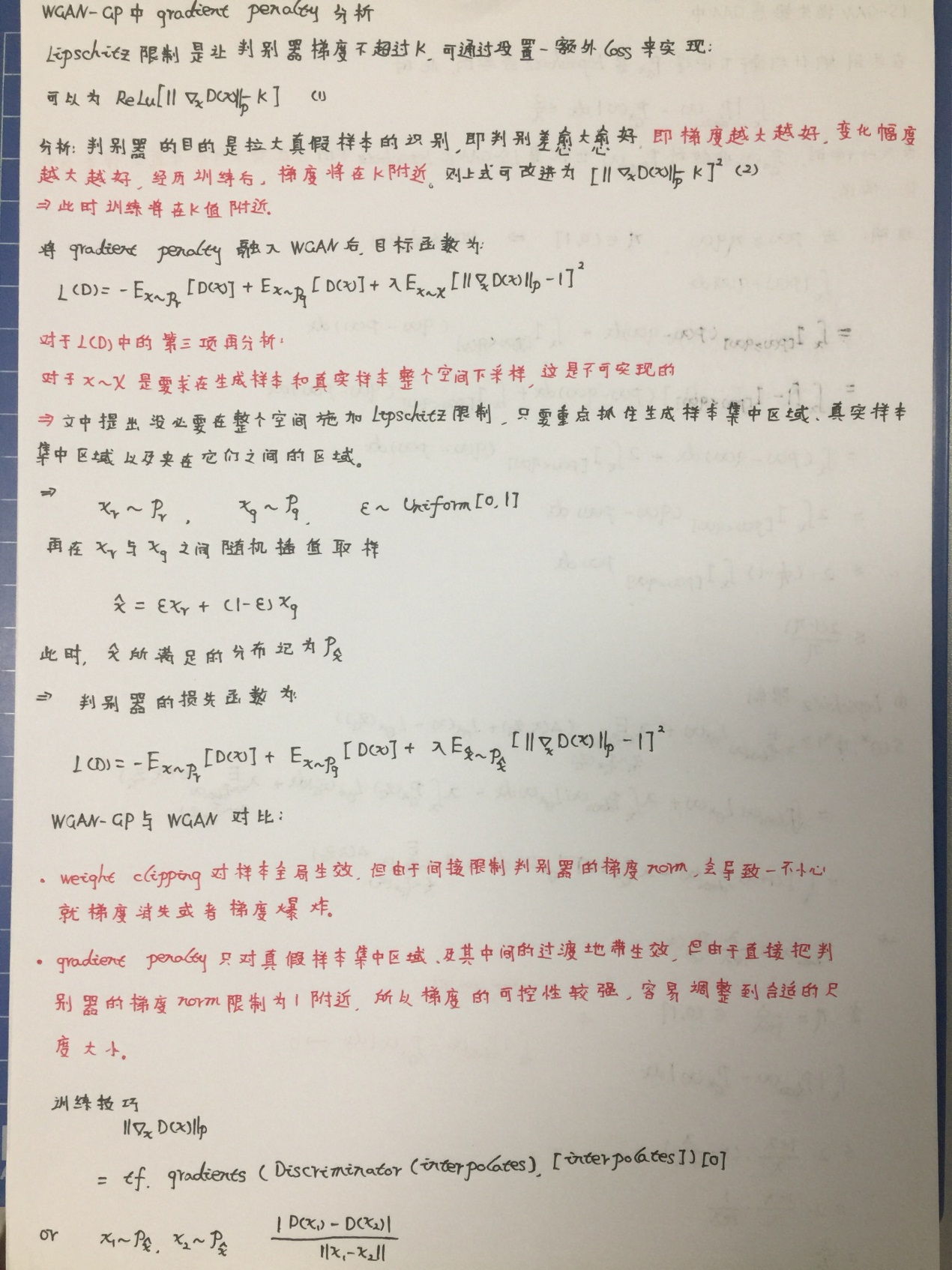
WGAN-GP的创新点也就在目标函数的第二项上，由于模型是对每个样本独立地施加梯度惩罚，所以判别器的模型架构中不能使用Batch Normalization，因为它会引入同个batch中不同样本的相互依赖关系。

小结：这周对LS-GAN的研读对文章的“按需分配”有了一定的理解，也在定量上分析了WGAN中的Lipschitz限制的作用。WGAN-GP指出了WGAN存在的两大问题，weight clipping导致的参数集中化和调参上的梯度爆炸和梯度消失问题，改进的gradient penalty解决了问题将参数与限制联系起来达到真实的Lipschitz限制条件。

本周实验室正式搬到新区了，有了硬件支持可以调调程序了，同时也要加强在编程上的能力。本周将CVAE-GAN的程序调了一下，在svhn的门牌号图像集上运行了一下效果还行，具体的程序我准备下周做个详细的分析并且更新到自己的github上。下周的时间把[Unsupervised Image-to-Image Translation Networks](https://arxiv.org/abs/1703.00848)研读一下，再整理一下近段时间的程序代码。

**附录：**

Lipschitz限制下生成样本与真实样本一致性证明

WGAN-GP中gradient penalty分析